

PARADIGMAS DE PROGRAMAÇÃO - RUBY

Alunos: Arthur Wallace Silva Lopes, César Tallys, Lucas Rodrigues Lopes

Instituto Federal de Brasília - Campus Taguatinga

Curso: Ciência da Computação

Disciplina: Paradigmas de Programação

Professor: Roberto Duarte Fontes

1. INTRODUÇÃO

A história e a teoria por trás do Ruby

O QUE É RUBY?



- Ruby é uma linguagem de programação de **alto nível**, ou seja, busca ter uma sintaxe mais próxima da fala humana e permite a criação de **programas complexos** de maneira relativamente **simples**.
- Ruby também possui uma **sintaxe intuitiva** e de fácil compreensão, o que a torna uma linguagem atraente para **iniciantes**. É uma linguagem de propósito geral, o que significa que pode ser utilizada em **diversos** tipos de **projetos**, incluindo aplicações web, aplicativos desktop e scripts de automação.

COMO O RUBY FOI CRIADO?

- A linguagem Ruby foi criada por Yukihiro Matsumoto, um programador e professor de informática japonês. Ele começou a trabalhar na criação da linguagem em 1993, com o objetivo de criar uma linguagem de programação que fosse mais **poderosa** do que as linguagens existentes, mas ao mesmo tempo fosse mais **simples** e mais **divertida** de se usar.
- Matz inspirou-se em várias linguagens de programação ao criar Ruby, como Smalltalk e Perl e também buscou incorporar alguns dos principais conceitos da filosofia do Zen Budismo, como **simplicidade**, **foco** e a importância da **experiência do usuário**.

POR QUE **RUBY** É IMPORTANTE?

Ruby é uma linguagem de programação importante por várias razões:

1. Ela é **amplamente utilizada** em projetos de tecnologia de todos os tipos, o que a torna uma linguagem valiosa para incluir no currículo de qualquer desenvolvedor;
2. Ela possui uma grande **comunidade** ativa e colaborativa, o que facilita o aprendizado e o compartilhamento de conhecimento. Isso a torna uma linguagem atraente para iniciantes e para aqueles que desejam se envolver com projetos open source;
3. Por último, Ruby é uma linguagem **versátil** e pode ser utilizada em muitos tipos diferentes de projetos.

2. CARACTERÍSTICAS

CARACTERÍSTICAS

▷ **Linguagem interpretada**

Ruby é uma linguagem interpretada, o que significa que o código é lido e executado diretamente pelo interpretador, sem precisar ser compilado previamente. Isso pode ser mais lento do que linguagens compiladas, mas também permite uma maior flexibilidade e facilidade de depuração.

▷ **Suporte a vários paradigmas**

Embora Ruby seja principalmente orientada a objetos, ela também suporta outros paradigmas de programação, como procedural e funcional. Isso permite a criação de programas mais flexíveis e adaptáveis.

CARACTERÍSTICAS

▷ **Ampla variedade de bibliotecas e frameworks**

A comunidade de Ruby desenvolveu uma ampla variedade de bibliotecas e frameworks que facilitam o desenvolvimento de aplicações em diversos campos. Alguns exemplos populares incluem o Rails para a criação de aplicações web, o Sinatra para a criação de APIs e o GTK+ para a criação de aplicativos desktop.

▷ **Ativa comunidade de desenvolvedores**

Ruby possui uma comunidade ativa e colaborativa de desenvolvedores, o que facilita o aprendizado e o compartilhamento de conhecimento sobre a linguagem. Além disso, a comunidade contribui para a manutenção e melhoria da linguagem, tornando-a uma opção estável e confiável para projetos de todos os tipos.

3. PLATAFORMA DE DESENVOLVIMENTO

PLATAFORMA DE DESENVOLVIMENTO

- ▷ A plataforma de desenvolvimento para Ruby inclui ambientes de desenvolvimento integrado (IDEs), terminais ou consoles e interpretadores. Alguns exemplos populares de IDEs para Ruby incluem o RubyMine, o Aptana Studio e o Eclipse.
- ▷ O terminal também pode ser utilizado para desenvolver aplicações em Ruby, sendo uma opção mais rápida e simples para projetos menores ou para aqueles que preferem trabalhar com linhas de comando.
- ▷ Por fim, o interpretador é o programa responsável por ler e executar o código Ruby. Existem várias opções de interpretadores disponíveis, como o MRI (Matz's Ruby Interpreter) e o JRuby, que é implementado em Java e pode ser executado em qualquer plataforma que tenha suporte a Java.

4. EJEMPLOS DE CÓDIGO

CLASSES E MÓDULOS

```
Classes e módulos

# Criação de uma classe
class ContaBancaria
  def initialize(numero, saldo)
    @numero = numero
    @saldo = saldo
  end

  def depositar(valor)
    @saldo += valor
  end

  def sacar(valor)
    @saldo -= valor
  end
end

# Criação de um módulo
module Imprimível
  def imprimir
    puts "Número da conta: #{@numero}"
    puts "Saldo: #{@saldo}"
  end
end

# Inclusão de um módulo em uma classe
class ContaPoupança < ContaBancaria
  include Imprimível

  def initialize(numero, saldo, taxa_juros)
    super(numero, saldo)
    @taxa_juros = taxa_juros
  end

  def aplicar_juros
    @saldo += @saldo * @taxa_juros
  end
end
```

Neste exemplo, criamos uma classe **ContaBancaria** que representa uma conta bancária genérica. Esta classe possui os métodos de instância **depositar** e **sacar**, que permitem fazer depósitos e saques na conta, respectivamente.

Em seguida, criamos um módulo **Imprimível** que possui um método imprimir, que exibe o número da conta e o saldo. Por fim, criamos a classe **ContaPoupança** que herda de **ContaBancaria** e inclui o módulo **Imprimível**. A classe **ContaPoupança** possui um método de instância adicional chamado **aplicar_juros**, que aplica juros à conta.

HERANÇA

```
Herança

# Criação de uma classe-base
class Veículo
  def inicializar(marca, modelo)
    @marca = marca
    @modelo = modelo
  end

  def ligar
    puts "Ligando o veículo..."
  end
end

# Criação de uma classe-filha que herda de
# Veículo
class Carro < Veículo
  def inicializar(marca, modelo, ano)
    super(marca, modelo)
    @ano = ano
  end

  def acelerar
    puts "Acelerando o carro..."
  end
end
```

Aqui, criamos uma classe chamada **Veículo** que possui dois métodos: **inicializar** e **ligar**. O método **inicializar** recebe dois argumentos, **marca** e **modelo**, e os atribui aos respectivos instance variables **@marca** e **@modelo**.

Em seguida, está sendo criada uma classe chamada **Carro** que herda de **Veículo**. Isso significa que **Carro** possui todos os métodos e atributos de **Veículo**, além de poder ter os seus próprios métodos e atributos.

NOVA INSTÂNCIA DE UMA CLASSE

```
Métodos de instância e de classe

class Pessoa
  def inicializar(nome, idade)
    @nome = nome
    @idade = idade
  end

  def cumprimentar
    puts "Olá, meu nome é #{@nome} e tenho #
    {@idade} anos."
  end
end

pessoa1 = Pessoa.new
pessoa1.inicializar("João", 30)
pessoa1.cumprimentar # Output: "Olá, meu nome é
João e tenho 30 anos."
```

Neste exemplo, criamos uma classe chamada **Pessoa** que possui um método de instância chamado **inicializar** e outro chamado **cumprimentar**.

Para instanciar um novo objeto baseado em uma classe, é utilizado a nomenclatura **new**. Dessa forma, após a inicialização podemos executar o método cumprimentar e observar a saída esperada.

5. PARADIGMAS ASSOCIADOS

PARADIGMAS

- Ruby é uma linguagem de programação que segue tanto o paradigma **orientado a objetos** como **funcional**, essa combinação permite ao desenvolvedor escolher a abordagem mais adequada para cada problema, podendo usar **heranças** e **encapsulamentos** quando é necessário e ainda trabalhar com **funções puras** e **imutabilidade** quando isso é vantajoso para a solução.

6.LEGIBILIDADE E CONFIABILIDADE

LEGIBILIDADE

- Ruby é uma linguagem com uma **sintaxe simples** e **intuitiva**, o que a torna **fácil** de **ler** e **entender**. Isso é importante para manter o código organizado e permitir que outras pessoas possam trabalhar no código com facilidade.
- Além disso, Ruby é uma linguagem de código aberto, o que significa que existe uma **grande comunidade** de desenvolvedores que contribuem com exemplos de código e documentação, tornando ainda mais fácil para os desenvolvedores entenderem e usarem a linguagem.

CONFIABILIDADE

- ▷ Ruby é conhecida por ser uma linguagem **confiável**. A linguagem conta com um sistema de tipos dinâmico, o que significa que os tipos de dados são determinados em tempo de execução, o que facilita o desenvolvimento, aumentando a produtividade e diminuindo a chance de erros.
- ▷ Além disso, Ruby conta com inúmeras bibliotecas e frameworks, como **Rails**, que oferecem uma grande variedade de funcionalidades, e garante **confiabilidade** e **estabilidade** no desenvolvimento de aplicações.

7. CUSTO

CUSTO

- ▷ Ruby é uma linguagem de programação de **código aberto**, o que significa que é **gratuita** e de **livre acesso** para qualquer pessoa. Isso significa que não há custos diretos para começar a usar a linguagem. No entanto, existem alguns custos indiretos que podem ser associados ao uso de Ruby como contratar desenvolvedores qualificados, infraestrutura e manutenção. É importante levar esses custos em consideração ao planejar o desenvolvimento de software com Ruby.

8. ONDE SE UTILIZA

ONDE SE UTILIZA

- ▶ Em geral, as empresas usam Ruby para desenvolver aplicações web, incluindo sites e aplicativos, como plataformas de comércio eletrônico, sistemas de gerenciamento de conteúdo, plataformas de gerenciamento de projetos e sistemas de atendimento ao cliente. Algumas empresas usam Ruby para automação e desenvolvimento de aplicativos desktop ou mobile.



airbnb



shopify



zendesk

9. MERCADO DE TRABALHO

MERCADO DE TRABALHO

- Ruby é uma linguagem de programação amplamente utilizada e tem uma **grande demanda** no mercado de trabalho já que muitas empresas utilizam Ruby para desenvolver aplicativos web.
- Isso inclui empresas de diferentes tamanhos e setores, como startups, empresas de tecnologia e empresas de negócios tradicionais. Como resultado, há uma grande demanda por desenvolvedores Ruby, especialmente para desenvolvimento web.

10. ASSOCIAÇÃO COM OS CURSOS DE COMPUTAÇÃO

ASSOCIAÇÃO COM OS CURSOS DE COMPUTAÇÃO

Ruby é uma linguagem de programação que pode ser relevante em cursos de Computação por várias razões:

- ▶ **Introdução à programação:** Ruby pode ser amplamente utilizado como uma linguagem de introdução à programação devido à sua **simplicidade** e **facilidade** de uso. É uma ótima opção para ensinar **conceitos básicos** de programação e **orientação a objetos** para estudantes iniciantes.
- ▶ **Desenvolvimento web:** Ruby é uma das principais linguagens de programação usadas para desenvolvimento web, especialmente com o uso do framework **Rails**. Como o desenvolvimento web é uma área em constante crescimento, é importante que os estudantes de computação tenham uma boa compreensão das tecnologias usadas nessa área.

11. DESENVOLVIMENTO DE APLICAÇÃO

API EM RUBY ON RAILS

Para exemplificar o desenvolvimento com Ruby, criamos uma API utilizando o framework Ruby on Rails

